Лабораторна робота №6
Дослідження можливостей виправлення помилок при кодуванні Хеммінга

Мета: Набути практичні навички застосування методів кодування з виправленням помилок за допомогою коду Хеммінга.

Обладнання та програмне забезпечення: персональний комп'ютер; будь-яка мова програмування; офісне програмне забезпечення для формування звітів та побудови діаграм.

Література

1. Соколов А. Теорія інформації та кодування. Лабораторний практикум [Електронний ресурс]: режим доступу: https://books.google.com.ua/books?id=XQRPDwAAQBAJ&printsec=copyright&redir_esc=y#v=onepage&q&f=false
2. Алгоритм Хеммінга. [Електронний ресурс]. - Режим доступу: https://en.wikipedia.org/wiki/Hamming_code
3. Євсеєв С.П. Кібербезпека: Основи кодування та криптографії / Євсеєв С.П., Мілов О.В., Остапов С.Е. Сєвєрінов О.В. - Харків, - Львів: Вид. ПП "Новий світ-2000", 2023. - 658 с.
4. Теорія інформації і кодування: курс лекцій [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за спеціальністю 124 «Системний аналіз» /; уклад.: А.Є.Коваленко. Київ : КПІ ім. Ігоря Сікорського, 2020. - 248 с.
5. Івашко А.В., Крилова В.А. Теорія інформації та кодування в прикладах і задачах: навч.-метод. посіб. Харків : НТУ «ХПІ», 2022. - 317 с.
6. Жураковський Ю.П., Полторак В.П. Теорія інформації та кодування: Підручник. – К.: Вища шк., 2001. – 255 с.

Теоретична частина

Код Хеммінга було розроблено у 1950 році Річардом Хеммінгом. Він належить до лінійних кодів і здатен виправляти однобітові та виявляти двобітові помилки при передаванні або зберігання даних. Сьогодні цей код використовується для контролю помилок у пам'яті (ECC) та при зберіганні даних , зокрема, у RAID2.

Для кращого розуміння того, як працює код Хеммінга, розглянемо приклад.

Нехай наші дані розміщено у табличці, кожна комірка якої — один біт інформації. Перший рядок містить номера бітів від 0 до 15. Третій рядок — ті ж номера бітів, записані у двійковій системі числення. Нульовий біт 0000 зарезервуємо для загального біта парності повідомлення. Він зафарбований сірим кольором. Біти, номера яких є ступенями двійки (2^0=1, 2^1=2, 2^2=4, 2^3=8 і т.д.) будемо вважати контрольними бітами. Їх особливість в тому, що вони мають лише одну одиничку у двійковому представленні. Вони зафарбовані блакитним кольором. Решту бітів (білих) вважатимемо інформаційними бітами.

Як бачимо, для контролю помилок потрібна деяка надмірність інформації, що надсилається/зберігається, оскільки для власне інформації з 16 бітів лишилося лише 11. У нашому випадку надмірність дорівнює 16/11 — 1 = 45%.

Нехай інформація, що надсилається, має вигляд: 10010011011. Запишемо її до таблички.

b0	*b1*	*b2*	*b3*	*b4*	*b5*	*b6*	*b7*	*b8*	*b9*	*b10*	*b11*	*b12*	*b13*	*b14*	*b15*
			1		0	0	1		0	0	1	1	0	1	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Тепер нам необхідно обчислити значення контрольних розрядів. Загальне правило таке: для обчислення значення контрольного розряду підсумовуємо значення усіх розрядів, двійковий номер яких містить “1” в тому ж розряді, що й контрольний. Отже для контрольного розряду $b1$ ми повинні підсумувати біти:

$$b1=b3\oplus b5\oplus b7\oplus b9\oplus b11\oplus b13\oplus b15=1\oplus 0\oplus 1\oplus 0\oplus 1\oplus 0\oplus 1=0.$$

Для інших контрольних розрядів отримаємо:

$$b2=b3\oplus b6\oplus b7\oplus b10\oplus b11\oplus b14\oplus b15=1\oplus 0\oplus 1\oplus 0\oplus 1\oplus 1\oplus 1=1;$$
$$b4=b5\oplus b6\oplus b7\oplus b12\oplus b13\oplus b14\oplus b15=0\oplus 0\oplus 1\oplus 1\oplus 0\oplus 1\oplus 1=0;$$
$$b8=b9\oplus b10\oplus b11\oplus b12\oplus b13\oplus b14\oplus b15=0\oplus 0\oplus 1\oplus 1\oplus 0\oplus 1\oplus 1=0.$$

Внесемо отримані значення до таблички. Отримаємо:

b0	*b1*	*b2*	*b3*	*b4*	*b5*	*b6*	*b7*	*b8*	*b9*	*b10*	*b11*	*b12*	*b13*	*b14*	*b15*
	0	**1**	1	**0**	0	0	1	**0**	0	0	1	1	0	1	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Тепер залишається обчислити загальний біт парності $b0$, який буде дорівнювати 0, якщо кількість одиниць у табличці парна, і 1, якщо ні. Таким чином, загальна кількість одиниць у цілому повідомлення має бути завжди парною. В нашому випадку кількість одиниць у бітах $b1$-$b15$ дорівнює семи, тобто непарна. Отже, значення біту парності має бути “1”. В результаті отримаємо:

b0	*b1*	*b2*	*b3*	*b4*	*b5*	*b6*	*b7*	*b8*	*b9*	*b10*	*b11*	*b12*	*b13*	*b14*	*b15*
1	**0**	**1**	1	**0**	0	0	1	**0**	0	0	1	1	0	1	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Результуюче повідомлення, яке ми надішлемо до отримувача, буде мати вигляд: **1011000100011011**.

На приймальному боці виконують майже такі обчислення, як і на передавальному.

Обчислюють значення так званого “синдрому”:

$$s0=b1\oplus b3\oplus b5\oplus b7\oplus b9\oplus b11\oplus b13\oplus b15=0\oplus 1\oplus 0\oplus 1\oplus 0\oplus 1\oplus 0\oplus 1=0;$$
$$s1=b2\oplus b3\oplus b6\oplus b7\oplus b10\oplus b11\oplus b14\oplus b15=1\oplus 1\oplus 0\oplus 1\oplus 0\oplus 1\oplus 1\oplus 1=0;$$
$$s2=b4\oplus b5\oplus b6\oplus b7\oplus b12\oplus b13\oplus b14\oplus b15=0\oplus 0\oplus 0\oplus 1\oplus 1\oplus 0\oplus 1\oplus 1=0;$$
$$s3=b8\oplus b9\oplus b10\oplus b11\oplus b12\oplus b13\oplus b14\oplus b15=0\oplus 0\oplus 0\oplus 1\oplus 1\oplus 0\oplus 1\oplus 1=0.$$
$$s3,s2,s1,s0=0,0,0,0=0.$$

Загальна парність повідомлення:

$$p = b0 \oplus b1 \oplus b2 \oplus b3 \oplus b4 \oplus b5 \oplus b6 \oplus b6 \oplus b7 \oplus b8 \oplus b9 \oplus b10 \oplus b11 \oplus b12 \oplus b13 \oplus b14 \oplus b15 = 0.$$

Таким чином, якщо синдром та парність дорівнюють нулю, то отримане повідомлення не зазнало змін при передаванні його каналами зв'язку.

Припустимо, що при передаванні даних у повідомлення спотворився один біт, наприклад, $b13$, замість нуля отримали одиницю:

b0	*b1*	*b2*	*b3*	*b4*	*b5*	*b6*	*b7*	*b8*	*b9*	*b10*	*b11*	*b12*	*b13*	*b14*	*b15*
1	**0**	**1**	1	**0**	0	0	1	**0**	0	0	1	1	1	1	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Обчислимо синдром та парність у цьому випадку:

$$s0 = b1 \oplus b3 \oplus b5 \oplus b7 \oplus b9 \oplus b11 \oplus b13 \oplus b15 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 1;$$
$$s1 = b2 \oplus b3 \oplus b6 \oplus b7 \oplus b10 \oplus b11 \oplus b14 \oplus b15 = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 0;$$
$$s2 = b4 \oplus b5 \oplus b6 \oplus b7 \oplus b12 \oplus b13 \oplus b14 \oplus b15 = 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1;$$
$$s3 = b8 \oplus b9 \oplus b10 \oplus b11 \oplus b12 \oplus b13 \oplus b14 \oplus b15 = 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1.$$
$$s3, s2, s1, s0 = 1, 1, 0, 1 = 1.$$

Парність:

$$p = b0 \oplus b1 \oplus b2 \oplus b3 \oplus b4 \oplus b5 \oplus b6 \oplus b6 \oplus b7 \oplus b8 \oplus b9 \oplus b10 \oplus b11 \oplus b12 \oplus b13 \oplus b14 \oplus b15 = 1.$$

Очевидно, парність та синдроми, відмінні від нуля, показують, що десь сталася помилка. В якому розряді вона сталася, - дозволяє визначити синдром s3,s2,s1,s0 = 1101. 1101 — це й є якраз двійковий номер біта, в якому сталася помилка — 1101 — біт $b13$. Достатньо його інвертувати і ми виправимо помилку. Ось так працює код виявлення та виправлення помилки Річарда Хеммінга.

Ми переконалися, що код Хеммінга дозволяє виправляти однобітові помилки. Однак, код Хеммінга може визначати наявність двобітних помилок, однак виправити їх вже не в змозі.

Продемонструємо це на нашому прикладі. Хай окрім 13-го біту ще було змінено 12-й. Тоді маємо:

b0	*b1*	*b2*	*b3*	*b4*	*b5*	*b6*	*b7*	*b8*	*b9*	*b10*	*b11*	*b12*	*b13*	*b14*	*b15*
1	**0**	**1**	1	**0**	0	0	1	**0**	0	0	1	0	1	1	1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

У цьому випадку синдром та парність будуть мати вигляд:

$$s0 = b1 \oplus b3 \oplus b5 \oplus b7 \oplus b9 \oplus b11 \oplus b13 \oplus b15 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 1;$$
$$s1 = b2 \oplus b3 \oplus b6 \oplus b7 \oplus b10 \oplus b11 \oplus b14 \oplus b15 = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 0;$$
$$s2 = b4 \oplus b5 \oplus b6 \oplus b7 \oplus b12 \oplus b13 \oplus b14 \oplus b15 = 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 0;$$
$$s3 = b8 \oplus b9 \oplus b10 \oplus b11 \oplus b12 \oplus b13 \oplus b14 \oplus b15 = 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 0.$$
$$s3, s2, s1, s0 = 0, 0, 0, 1 = 1.$$

Парність:

$$p = b0 \oplus b1 \oplus b2 \oplus b3 \oplus b4 \oplus b5 \oplus b6 \oplus b6 \oplus b7 \oplus b8 \oplus b9 \oplus b10 \oplus b11 \oplus b12 \oplus b13 \oplus b14 \oplus b15 = 0.$$

Бачимо, що в цьому випадку, на відміну від попереднього, коли була однобітова помилка, синдром та парність не збігаються, що й сигналізує про парну помилку. Однак виправити її ми не можемо, оскільки не знаємо правильних адрес комірок. Отже, двобітові помилки код Хеммінга виправити не може.

Детальніше дізнатися про метод кодування за Хеммінгом можна з літературних джерел до цієї лабораторної роботи, зокрема [2].

Практична частина

Для виконання цієї лабораторної роботи необхідно зробити таке:

1. Складіть програму (будь-якою мовою програмування), яка кодує/декодує довільне вхідне текстове повідомлення методом Хеммінга та виводить результат на екран/файл.
2. Обчисліть коефіцієнт надмірності для 2-3 різних повідомлень. Поясніть отримані результати.
3. Підготуйте звіт з лабораторної роботи, до якого включіть:

- Поясніть алгоритм Хеммінга, його переваги та недоліки. Наведіть приклади на основі Ваших обчислень у цій лабораторній роботі.
- Обчисліть середній коефіцієнт надмірності по результатах Ваших досліджень.
- Дайте відповіді на контрольні запитання.

Контрольні запитання

1. Поясніть особливості алгоритму Хеммінга.
2. Де використовується алгоритм Хеммінга сьогодні?
3. Які переваги та недоліки алгоритму Хеммінга Ви бачите?
4. Порівняйте результати Ваших досліджень зі іншими методами виправлення помилок, які Ви знаєте.